Materialprüflabor

Aufgaben

In einem Materialprüflabor werden unterschiedliche Stoffe auf ihre Eigenschaften, wie Dicke, Zugfestigkeit, Gewicht und anderes von Mitarbeiterinnen und Mitarbeitern (im Folgenden Mitarbeiter genannt) geprüft.

1 Objektorientierte Entwicklung des MAPD (Mess-Auswert-Protokoll-und-Dokumentationsprogramm). Ein erstes UML-Klassendiagramm finden Sie in Material 1.

1.1 Beschreiben Sie das gegebene Klassendiagramm.

Hinweis: Die Stichworte Klasse, Objekt, Attribut, Methode, Beziehung, Multiplizität und Rolle sind sinnvoll zu verwenden.

(5 BE)

1.2 In Material 2 ist exemplarisch ein Prüfprotokoll dargestellt. Zeichnen Sie ein UML-Objektdiagramm auf Grundlage dieses Protokolls unter Berücksichtigung des Klassendiagramms in Material 1.

Hinweis: Fehlende Attributwerte sind durch sinnvolle Werte zu ergänzen.

(7 BE)

1.3 Überführen und implementieren Sie die Klasse `Probe` in einer objektorientierten Programmiersprache.

Hinweise: `setzeTester()`, `erstelleProtokoll()` sowie `erfasseMesswert()` der Klasse `Probe` sind in Material 1 beschrieben. Die Dokumentationen der Klasse `List` ist in Material 3 zu finden.

(8 BE)

1.4 Auf einen Kundenwunsch hin soll ein Median-Wert in die Statistik integriert werden.
Der Median wird ermittelt, indem man die Messwerte der Größe nach aufsteigend sortiert. Ist es eine ungerade Anzahl von Messwerten, ist der Median genau in der Mitte. Ist die Anzahl der Messwerte gerade, wird der Median als Mittelwert aus den beiden mittleren Messwerten errechnet.

1.4.1 Implementieren Sie die Methode `ermittleMedian()` für die Klasse `Messwertsatz`.

(7 BE)

1.4.2 Als Alternative kann die Methode `hinzufuegenWert()` der Klasse `Messwertsatz` so erweitert werden, dass jeder Messwert nach seinem Attribut `wert` sortiert eingefügt wird sowie die Attribute `min` und `max` aktualisiert werden. Implementieren Sie diese Funktionalität und beschreiben Sie die an der Methode `ermittleMedian()` vorzunehmenden Anpassungen nach der Implementierung der Methode `hinzufuegenWert()`.

(9 BE)

1.5 In einer Messstation können unterschiedliche Messwerkzeuge verwendet werden. Das verwendete Messwerkzeug bestimmt dabei die Art, wie eine Messung durchgeführt wird, und die Form, wie die gewonnenen Ergebnisse vom Messwerkzeug über eine serielle Schnittstelle an die Messtation übermittelt werden. Für diese Unterscheidung besitzt der Konstruktor der Klasse `Messtation` den Parameter `messart`.

- Wird der Parameter `messart` mit dem Wert `0` initialisiert, wird nur eine Zeile mit dem Resultattext von der seriellen Schnittstelle übertragen. In dieser Zeile wird zuerst der Messwert, danach ein Leerzeichen, gefolgt von der gemessenen Einheit übertragen.
- Wird der Parameter `messart` hingegen mit dem Wert `1` initialisiert, werden die Messwerte einer Zugprüfmaschine mittels der Methode `holeKraft()` erfasst und ausgewertet. Eine Messung wird dabei so lange durchgeführt, bis einer der übermittelten Messwerte zehn Prozent kleiner ist als das bisherige gemessene Maximum.
- Als Endzeichen wird das Steuerzeichen `EOT` verwendet.
- Die Messstation antwortet mit dem Steuerzeichen `ACK`, wenn der Messwert korrekt übertragen wurde. Andernfalls wird mit dem Steuerzeichen `NAK` geantwortet.

1.5.1 Es werden pro Messung nacheinander vier Byte für die Kraftmessung und ein Byte für das Endzeichen übertragen. Die Werte der Messungen werden einzeln als Byte übertragen und einem Array der aktuell gemessenen Messwerte hinzugefügt. Eine Messung ist dann zu werten, wenn alle Bytes empfangen werden konnten. Diese ergeben die Messung, welche konvertiert zurückgegeben wird.
Entwickeln und zeichnen Sie für die Methode `holeKraft()` ein Struktogramm.

Hinweis: Zur Konvertierung einer Messung in eine Double-Zahl steht die Methode `konvertiereMesswert()`der Klasse `Messstation` Verfügung, die in Material 1 beschrieben ist.

(6 BE)

1.5.2 Implementieren Sie die Methode `erwarteMessung()` der Klasse `Messstation` für beide Messarten.

Hinweise: Die Dokumentationen der Klassen `DateTime`, `List` und `String` sind in Material 3 zu finden. Zur Umwandlung eines Strings in eine Double-Zahl steht die Methode `Double.parseDouble(s:String):double` zur Verfügung.

(8 BE)

1.6 Die Methode `starteMessreihe()` der Klasse `Messstation` soll für eine Probe insgesamt fünf Messwerte für die Materialeigenschaft Dicke erfassen. Dazu wird die entsprechende Probe gesucht (`sucheProbe()`). Anschließend wird die Messung erwartet und ein einzelner Messwert wird erzeugt und den bereits existierenden Messwerten hinzugefügt.
Entwickeln und zeichnen Sie ein Sequenzdiagramm für die Methode.

Hinweise: Eine Vorlage ist in Material 4 zu finden. Es ist davon auszugehen, dass eine entsprechende Probe mit fünf Messwerten bereits existiert.

(10 BE)

2 Entwicklung einer Datenbank für das Materialprüflabor. Ein erstes Entity-Relationship-Modell (ERM) ist in Material 5 dargestellt.

2.1 Beschreiben Sie den Aufbau und die Notationselemente des dargestellten ERM.

Hinweis: Die Stichworte Entitätstyp, Attribut, Primärschlüssel und Kardinalitäten sind sinnvoll zu verwenden.

(5 BE)

2.2 Überführen Sie das ERM in das relationale Modell in der 3. Normalform und begründen Sie Ihre Vorgehensweise.

Hinweis: Alle Relationen sind in der Schreibweise `Relation(PK, Attribut, ..., FK#)` anzugeben.

(8 BE)

2.3 Im Material 6 ist ein Tagesablauf im Materialprüflabor dargestellt. Geben Sie die in dieser Tabelle auftretenden Redundanzen an und beschreiben Sie mögliche Anomalien anhand von Beispielen.

(4 BE)

2.4 Einige Daten der Prüfdatenbank sollen ergänzt, geändert und ausgewertet werden.

2.4.1 Die Einheit Rho mit der Bezeichnung Massendichte soll gelöscht werden.
Formulieren Sie die entsprechenden SQL-Anweisung.

(3 BE)

2.4.2 Entwickeln Sie eine SQL-Anweisung, die die Anzahl aller Produkte pro Farbe auflistet, von denen mehr als fünf Produkte die identische Farbe besitzen. Die Ausgabe soll nach der Anzahl der Produkte absteigend sortiert werden.

(5 BE)

2.4.3 Mitarbeiter, die mehr als 500 Chargen geprüft haben, sollen ausgezeichnet werden. Entwickeln Sie eine SQL-Anweisung, die für die entsprechenden Mitarbeiter die Anzahl der geprüften Chargen ausgibt. Die Ausgabe soll nach der Anzahl der Chargen absteigend sortiert werden.

(5 BE)

2.4.4 Entwickeln Sie eine SQL-Anweisung, die den Hersteller und die Bezeichnung aller Prüfgeräte ausgibt, die nicht zum Prüfen benutzt wurden.

(3 BE)

2.5 Die Datenbank soll weitere Anforderungen umsetzen:

- Jede Charge eines Produkts muss mindestens einer Prüfung unterzogen werden.
- Zur Qualitätssicherung ist es wichtig, dass für jede Prüfung nachvollziehbar ist, welcher Mitarbeiter die Prüfung durchgeführt hat, welche Messwerte erzeugt werden konnten und nach welcher ISO-Prüfnorm die Prüfung durchgeführt wurde.
- Die Vollständigkeit dieser Daten wird durch einen QM-Mitarbeiter bestätigt.
- Jeder QM-Mitarbeiter ist ebenfalls Mitarbeiter, nicht alle Mitarbeiter sind QM-Mitarbeiter.

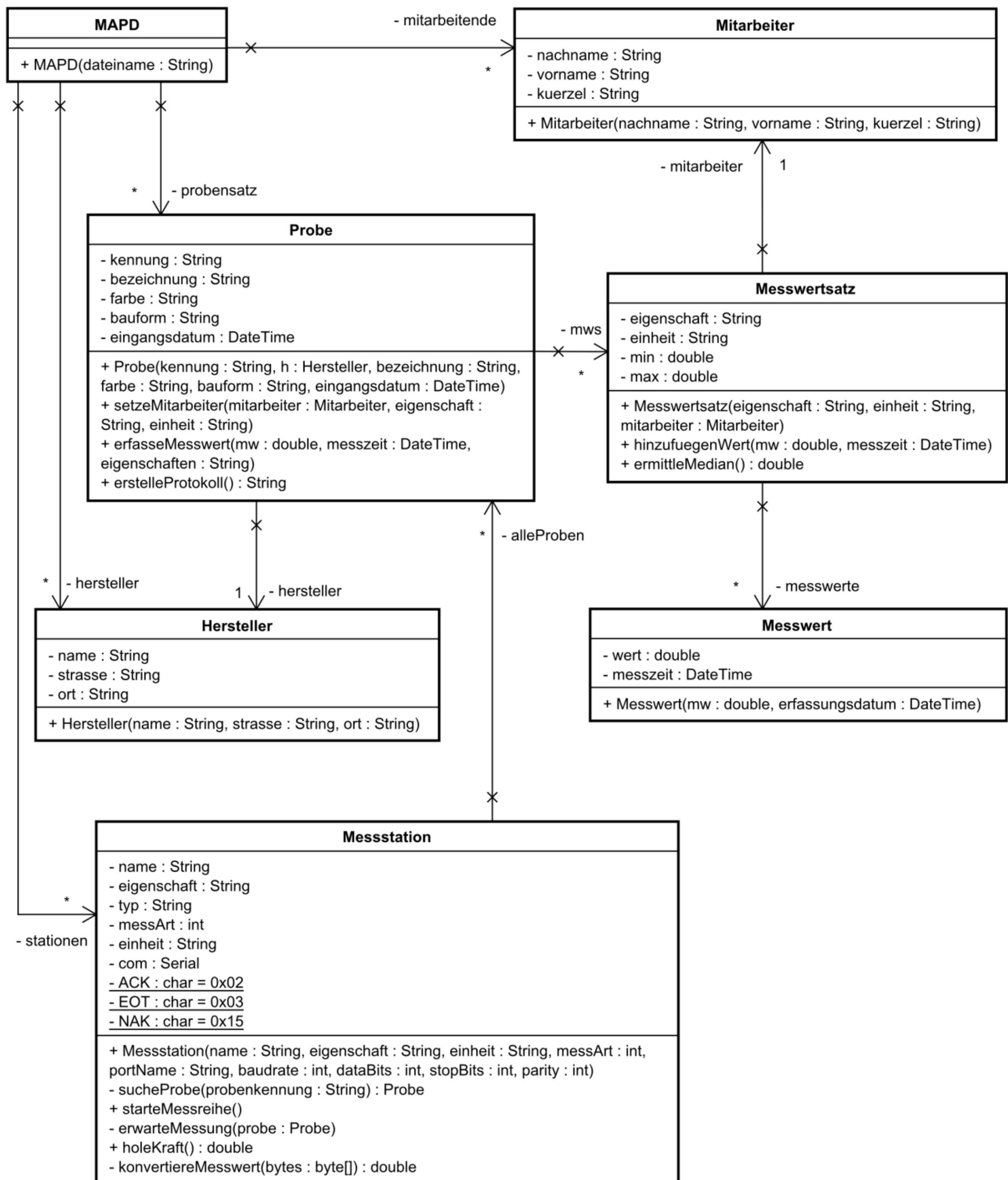
Entwickeln und zeichnen Sie die Erweiterung des ERM aus Material 5, das die genannten Anforderungen umsetzt.

Hinweise: Jede Prüfung findet nach genau einer Norm statt. Machen Sie deutlich, an welcher Stelle des alten Modells Sie neue Entitätstypen einbinden und stellen Sie die Relationen inklusive [min, max]-Notation dar. Attribute müssen nicht abgebildet werden.

(7 BE)

Material 1

UML-Klassendiagramm: MAPD



Material 1 (Fortsetzung)

Klasse `Probe`

- `setzeMitarbeiter()` prüft, ob das Attribut `eigenschaft` schon in einem bestehenden Messwertsatz vorliegt. Ist dies der Fall, wird der Mitarbeiter aktualisiert. Alternativ wird ein neuer Messwertsatz erzeugt.
- `erstelleProtokoll()` ergibt eine Zeichenkette der Form:

```
Kennung: PAP14783
Hersteller: Papierwerk Aschaffenburg
Bezeichnung: Universalpappe Robust
Eigenschaft: Dicke
Einheit: mm|Min: 1.0|Max: 10.0
Eigenschaft: Gewicht
Einheit: g/m|Min: 11.0|Max: 30.0
Eigenschaft: Berstdruck
Einheit: MPa|Min: 4.0|Max: 5.0
```

- `erfasseMesswert()` sucht über das Attribut `eigenschaft` den zugehörigen Messwertsatz und fügt diesem die Werte hinzu.

Klasse `Messtation`

- `konvertiereMesswert()` nimmt ein Array mit 4 Bytes als Parameter entgegen und liefert den Messwert als Double zurück.

Auf alle Attribute kann mittels set- und get-Methoden zugegriffen werden.

Material 2

Prüfprotokoll

Im Rahmen der Qualitätskontrolle wurde am 14.05.2023 eine neue Probe mit der Kennung PAP14783, der Bezeichnung Universalpappe Robust, mit der Farbe Grau und der Bauform C-Welle vom Hersteller Papierwerk, mit Sitz in der Papierwerkstraße 1 in 63777 Aschaffenburg aufgenommen.

Folgende Daten wurden dabei ermittelt:

Eigenschaft	Einheit	Minimum	Maximum	Anzahl Messwerte	Mitarbeiter
Gewicht	g/m²	295.3	304.7	3	Beate Salari
Berstdruck	mpa	100.5	240.5	2	Beate Salari

Messwerte Gewicht	
Messzeit	**Wert**
14.05.2023 10:00:05	295,3
14.05.2023 11:00:05	304,7
14.05.2023 12:00:05	300,0

Messwerte Berstdruck	
Messzeit	**Wert**
14.05.2023 10:10:04	240,5
14.05.2023 12:09:45	100,5

Material 3

Klassendokumentationen

Klasse `DateTime`

`DateTime()`
erzeugt ein `DateTime`-Objekt mit dem aktuellen Systemdatum und der Systemuhrzeit, im Format `dd.mm.yyyy hh:mm`.

`toString(): String`
liefert eine String-Repräsentanz des `Date`-Objekts im Format "`dd.mm.yyyy`".

DateTime
+ DateTime() + toString() : String

Klasse `String`

`equals(str: String): boolean`
liefert `true`, wenn beide Strings gleich sind, andernfalls `false`.

`split(str: String): String[]`
teilt einen String am Trennzeichen `str`. Die Teil-Strings werden in einem Feld zurückgeliefert.

`length(): int`
liefert die Anzahl der Zeichen des Strings.

String
+ equals(str : String) : boolean + split(str : String) : String[ ] + length() : int

Klasse `List`

Mithilfe der Klasse `List` können Objekte im Arbeitsspeicher in einer linearen Liste verwaltet werden. `List` repräsentiert eine generische Liste mit Elementen des Typs `T` in festgelegter Reihenfolge, auf die sowohl wahlfrei als auch sequenziell zugegriffen werden kann.

`List<T>()`
erzeugt eine generische Liste mit Elementen des Typs `T`.

`add(obj: T)`
hängt das Objekt `obj` vom Typ `T` am Ende der Liste an.

`add(index: int, obj: T)`
hängt das Objekt `obj` vom Typ `T` an der Position `index` an.

`contains(obj: T): boolean`
liefert `true`, wenn das Objekt `obj` in der Liste enthalten ist, sonst `false`.

`get(index: int): T`
liefert das Listenelement an der Position `index` zurück bzw. `null`, falls `index` negativ oder größer gleich der Anzahl der momentan enthaltenen Elemente ist.

`remove(index: int): T`
entfernt das Listenelement an der Position `index`. Liefert das entfernte Element zurück bzw. `null`, falls `index` negativ oder größer gleich der Anzahl der momentan enthaltenen Elemente ist.

`remove(obj: T): boolean`
entfernt das Objekt `obj` aus der Liste. Falls `obj` mehrmals in der Liste enthalten ist, wird nur das erste Vorkommen entfernt. Der Rückgabewert ist `true`, falls das Objekt gefunden und entfernt wurde, sonst `false`.

`size(): int`
liefert die Anzahl der Elemente in der Liste zurück.

List<T>
+ List<T>() + add(obj : T) + add(index : int, obj : T) + contains(obj : T) : boolean + get(index : int) : T + remove(index : int) : T + remove(obj : T) : boolean + size() : int

Klasse `Serial`

Serial
+ open() : boolean + read() : byte + write(value : char)

Ein Exemplar der Klasse Serial ermöglicht die Kommunikation über die serielle Schnittstelle.

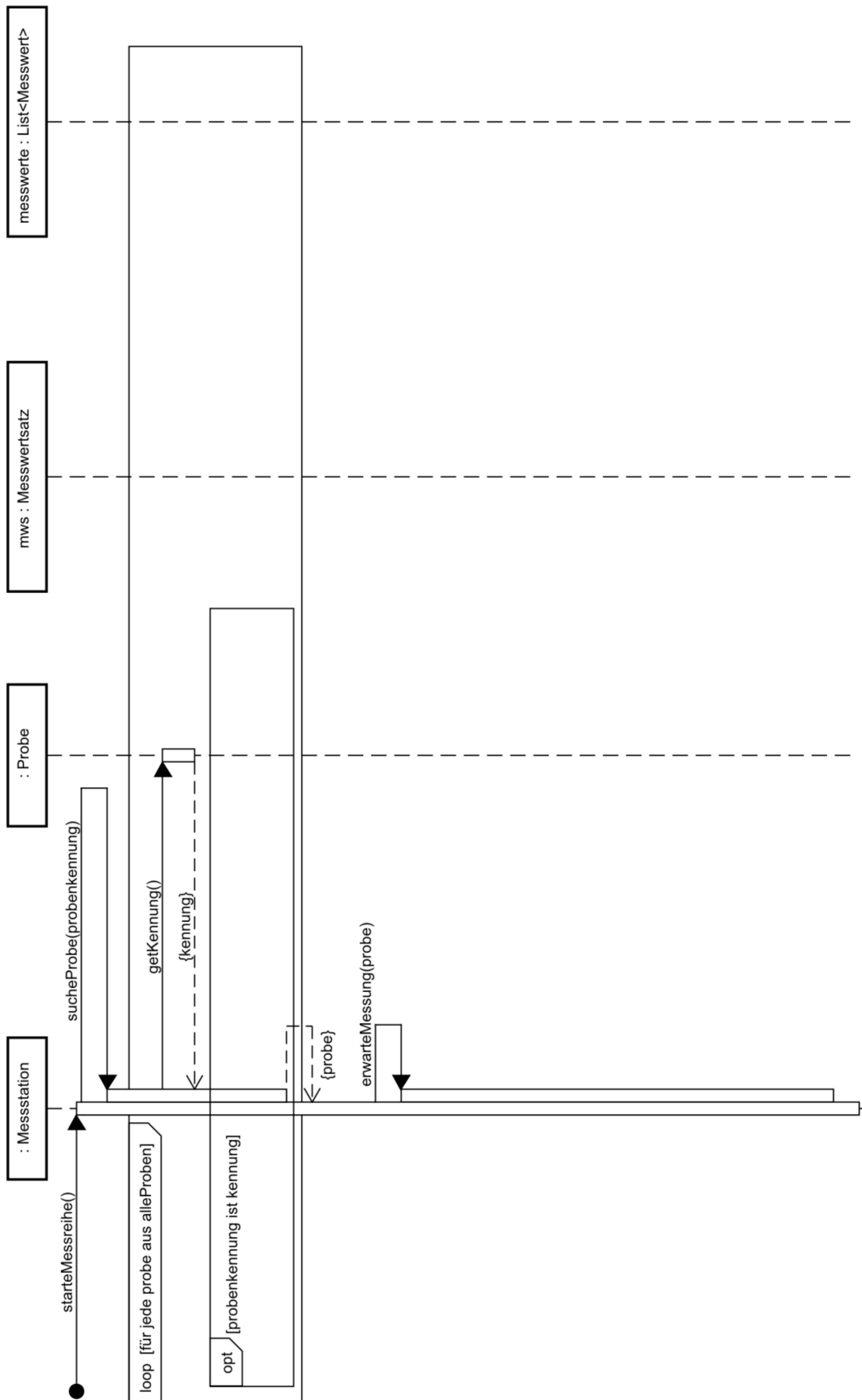
`open()` öffnet die serielle Schnittstelle; liefert `true`, wenn die Schnittstelle verwendbar ist.

`read()` liest ein Byte (0..255) von der serielle Schnittstelle, bzw. –1, wenn die Schnittstelle nicht geöffnet ist. Die Methode blockiert, bis ein Byte verfügbar ist.

`write()` schreibt ein `char` auf die serielle Schnittstelle; ist die Schnittstelle nicht geöffnet geschieht nichts.

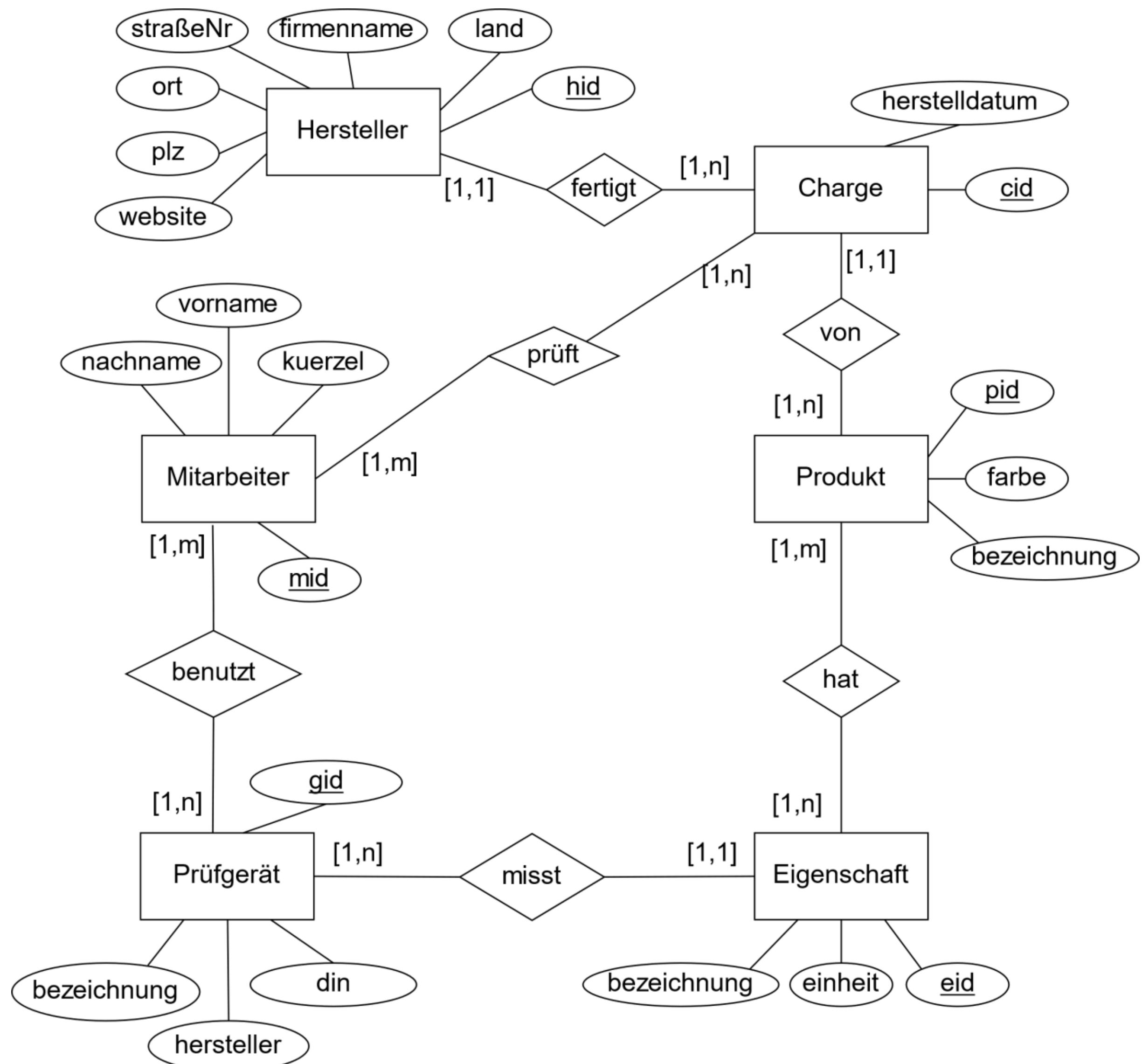
Material 4

Vorlage UML-Sequenzdiagramm



Material 5

Entity-Relationship-Modell



Material 6

Tabelle Tagesablauf Prüflabor (Ausschnitt)

Probennummer	Dicke	Berstdruck	Gewicht	Mitarbeiter	Hersteller
2203123	0,45 4,5 4,45	234,56 234,65 222	123 134 112	MAMU MU MA	PWA Aschaffenb.
2203123	5.55 5.53 5,55	400 401 404	128 127 129	MA MA MA	HWP
2203021	3.4 3.5	Prüfgerät defekt	100, 102, 104	PR PT PT	HMP
2203121	44 43 44	120 112 121	123 120 123	MO PT RR	PWA Aschaffenburg